HEXAWARE

Coding Challenge1

1. Provide a SQL script that initializes the database for the Job Board scenario "CareerHub".
2. Create tables for Companies, Jobs, Applicants and Applications.
3. Define appropriate primary keys, foreign keys, and constraints.
4. Ensure the script handles potential errors, such as if the database or tables already exist.

```
mysql> create database CareerHub;
Query OK, 1 row affected (0.02 sec)

mysql> use CareerHub;
Database changed
mysql> CREATE TABLE Companies (
    ->     CompanyID INT PRIMARY KEY,
    ->     CompanyName VARCHAR(255),
    ->     Location VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Jobs (
    ->     JobID INT PRIMARY KEY,
    ->     CompanyID INT,
    ->     JobTitle VARCHAR(255),
    ->     JobDescription TEXT,
    ->     JobLocation VARCHAR(255),
    ->     Salary DECIMAL,
    ->     JobType VARCHAR(255),
    ->     PostedDate DATETIME,
    ->     FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE Applicants (
    ->     ApplicantID INT PRIMARY KEY,
    ->     FirstName VARCHAR(255),
    ->     LastName VARCHAR(255),
    ->     Email VARCHAR(255),
    ->     Phone VARCHAR(20),
    ->     Resume TEXT
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE Applications (
    ->     ApplicationID INT PRIMARY KEY,
    ->     JobID INT,
    ->     ApplicantID INT,
    ->     ApplicationDate DATETIME,
    ->     CoverLetter TEXT,
    ->     FOREIGN KEY (JobID) REFERENCES Jobs(JobID),
    ->     FOREIGN KEY (ApplicantID) REFERENCES Applicants(ApplicantID)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> INSERT INTO Companies (CompanyID, CompanyName, Location) VALUES
    -> (1, 'TechCorp', 'Silicon Valley'),
    -> (2, 'FinanceCo', 'New York'),
    -> (3, 'MedTech Solutions', 'Boston'),
    -> (4, 'Data Innovators', 'San Francisco'),
    -> (5, 'Green Energy Solutions', 'Los Angeles'),
    -> (6, 'CyberSec Guardians', 'Washington, D.C.'),
    -> (7, 'HealthHub Inc.', 'Chicago'),
    -> (8, 'Manufacturing Innovations', 'Detroit'),
    -> (9, 'EducationTech Systems', 'Austin'),
    -> (10, 'E-commerce Express', 'Seattle');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> -- Insert Example Data into Jobs Table
mysql> INSERT INTO Jobs (JobID, CompanyID, JobTitle, JobDescription, JobLocation, Salary, JobType, PostedDate) VALUES
    -> (101, 1, 'Software Engineer', 'Developing cutting-edge software solutions.', 'Silicon Valley', 120000.00, 'Full-time', '2023-01-01 10:00:00'),
    -> (102, 2, 'Financial Analyst', 'Analyzing financial data and preparing reports.', 'New York', 90000.00, 'Full-time', '2023-01-02 11:30:00'),
    -> (103, 3, 'Medical Researcher', 'Conducting research on innovative medical technologies.', 'Boston', 110000.00, 'Full-time', '2023-01-03 12:45:00'),
    -> (104, 4, 'Data Scientist', 'Building machine learning models for data analysis.', 'San Francisco', 130000.00, 'Full-time', '2023-01-04 09:15:00'),
    -> (105, 5, 'Environmental Engineer', 'Designing eco-friendly energy solutions.', 'Los Angeles', 95000.00, 'Full-time', '2023-01-05 14:00:00'),
    -> (106, 6, 'Cybersecurity Specialist', 'Securing networks and systems from cyber threats.', 'Washington, D.C.', 115000.00, 'Full-time', '2023-01-06 13:20:00'),
    -> (107, 7, 'Healthcare IT Consultant', 'Implementing technology solutions in healthcare.', 'Chicago', 100000.00, 'Full-time', '2023-01-07 15:45:00'),
    -> (108, 8, 'Manufacturing Engineer', 'Optimizing production processes in manufacturing.', 'Detroit', 105000.00, 'Full-time', '2023-01-08 16:30:00'),
    -> (109, 9, 'EdTech Content Developer', 'Creating educational content for online learning platforms.', 'Austin', 85000.00, 'Full-time', '2023-01-09 08:45:00'),
    -> (110, 10, 'E-commerce Manager', 'Overseeing online sales and marketing operations.', 'Seattle', 120000.00, 'Full-time', '2023-01-10 17:00:00');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> -- Insert Example Data into Applicants Table
mysql> INSERT INTO Applicants (ApplicantID, FirstName, LastName, Email, Phone, Resume) VALUES
    -> (1, 'John', 'Doe', 'john.doe@example.com', '+1234567890', 'John_Doe_Resume.pdf'),
    -> (2, 'Jane', 'Smith', 'jane.smith@example.com', '+9876543210', 'Jane_Smith_Resume.docx'),
    -> (3, 'Michael', 'Johnson', 'michael.johnson@example.com', '+1112223333', 'Michael_Johnson_Resume.txt'),
    -> (4, 'Emily', 'Williams', 'emily.williams@example.com', '+4445556666', 'Emily_Williams_Resume.pdf'),
    -> (5, 'Daniel', 'Brown', 'daniel.brown@example.com', '+7778889999', 'Daniel_Brown_Resume.docx'),
    -> (6, 'Sophia', 'Miller', 'sophia.miller@example.com', '+3332221111', 'Sophia_Miller_Resume.txt'),
    -> (7, 'Matthew', 'Davis', 'matthew.davis@example.com', '+6667778888', 'Matthew_Davis_Resume.pdf'),
    -> (8, 'Olivia', 'Martinez', 'olivia.martinez@example.com', '+9998887777', 'Olivia_Martinez_Resume.docx'),
    -> (9, 'Christopher', 'Taylor', 'christopher.taylor@example.com', '+5554443333', 'Christopher_Taylor_Resume.txt'),
    -> (10, 'Ava', 'Jackson', 'ava.jackson@example.com', '+22233344444', 'Ava_Jackson_Resume.pdf');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> -- Insert Example Data into Applications Table
mysql> INSERT INTO Applications (ApplicationID, JobID, ApplicantID, ApplicationDate, CoverLetter) VALUES
    -> (1, 101, 1, '2023-02-01 08:30:00', 'I am excited about the Software Engineer position at TechCorp.'),
    -> (2, 102, 2, '2023-02-02 10:45:00', 'I am applying for the Financial Analyst role at FinanceCo.'),
    -> (3, 103, 3, '2023-02-03 12:15:00', 'I am passionate about medical research and am applying for the position at MedTech Solutions.'),
    -> (4, 104, 4, '2023-02-04 09:00:00', 'I am a data scientist with experience in machine learning. Excited about the opportunity at Data Innovators.'),
    -> (5, 105, 5, '2023-02-05 14:30:00', 'I am applying for the Environmental Engineer role at Green Energy Solutions.'),
    -> (6, 106, 6, '2023-02-06 13:00:00', 'As a cybersecurity specialist, I am interested in joining CyberSec Guardians.'),
    -> (7, 107, 7, '2023-02-07 15:30:00', 'I am applying for the Healthcare IT Consultant position at HealthHub Inc.'),
    -> (8, 108, 8, '2023-02-08 16:45:00', 'I am a manufacturing engineer seeking opportunities at Manufacturing Innovations.'),
    -> (9, 109, 9, '2023-02-09 08:15:00', 'I am interested in the EdTech Content Developer role at EducationTech Systems.'),
    -> (10, 110, 10, '2023-02-10 17:15:00', 'I am applying for the E-commerce Manager position at E-commerce Express.');
```

5. Write an SQL query to count the number of applications received for each job listing in the "Jobs" table. Display the job title and the corresponding application count. Ensure that it lists all jobs, even if they have no applications.

```
mysql> SELECT
    ->     Jobs.JobID,
    ->     Jobs.JobTitle,
    ->     COUNT(Applications.ApplicationID) AS ApplicationCount
    -> FROM
    ->     Jobs
    -> LEFT JOIN
    ->     Applications ON Jobs.JobID = Applications.JobID
    -> GROUP BY
    ->     Jobs.JobID, Jobs.JobTitle;
+-------+---------------------------+-------------------+
| JobID | JobTitle                  | ApplicationCount  |
+-------+---------------------------+-------------------+
|   101 | Software Engineer         |                 1 |
|   102 | Financial Analyst         |                 1 |
|   103 | Medical Researcher        |                 1 |
|   104 | Data Scientist            |                 1 |
|   105 | Environmental Engineer    |                 1 |
|   106 | Cybersecurity Specialist  |                 1 |
|   107 | Healthcare IT Consultant  |                 1 |
|   108 | Manufacturing Engineer    |                 1 |
|   109 | EdTech Content Developer  |                 1 |
|   110 | E-commerce Manager        |                 1 |
+-------+---------------------------+-------------------+
10 rows in set (0.00 sec)
```

6. Develop an SQL query that retrieves job listings from the "Jobs" table within a specified salary range. Allow parameters for the minimum and maximum salary values. Display the job title, company name, location, and salary for each matching job.

```
mysql> SELECT
    ->     JobTitle,
    ->     (SELECT CompanyName FROM Companies WHERE Companies.CompanyID = Jobs.CompanyID) AS CompanyName,
    ->     JobLocation,
    ->     Salary
    -> FROM
    ->     Jobs
    -> WHERE
    ->     Salary BETWEEN 95000 AND 125000;
+--------------------------+--------------------------+------------------+--------+
| JobTitle                 | CompanyName              | JobLocation      | Salary |
+--------------------------+--------------------------+------------------+--------+
| Software Engineer        | TechCorp                 | Silicon Valley   | 120000 |
| Medical Researcher       | MedTech Solutions        | Boston           | 110000 |
| Environmental Engineer   | Green Energy Solutions   | Los Angeles      |  95000 |
| Cybersecurity Specialist | CyberSec Guardians       | Washington, D.C. | 115000 |
| Healthcare IT Consultant | HealthHub Inc.           | Chicago          | 100000 |
| Manufacturing Engineer   | Manufacturing Innovations | Detroit          | 105000 |
| E-commerce Manager       | E-commerce Express       | Seattle          | 120000 |
+--------------------------+--------------------------+------------------+--------+
7 rows in set (0.00 sec)
```

7. Write an SQL query that retrieves the job application history for a specific applicant. Allow a parameter for the ApplicantID, and return a result set with the job titles, company names, and application dates for all the jobs the applicant has applied to.

```
mysql> SELECT
    ->     Jobs.JobTitle,
    ->     Companies.CompanyName,
    ->     Applications.ApplicationDate
    -> FROM
    ->     Applications
    -> JOIN
    ->     Jobs ON Applications.JobID = Jobs.JobID
    -> JOIN
    ->     Companies ON Jobs.CompanyID = Companies.CompanyID
    -> WHERE
    ->     Applications.ApplicantID = ApplicantID
    -> ;
+--------------------------+--------------------------+---------------------+
| JobTitle                 | CompanyName              | ApplicationDate     |
+--------------------------+--------------------------+---------------------+
| Software Engineer        | TechCorp                 | 2023-02-01 08:30:00 |
| Financial Analyst        | FinanceCo                | 2023-02-02 10:45:00 |
| Medical Researcher       | MedTech Solutions        | 2023-02-03 12:15:00 |
| Data Scientist           | Data Innovators          | 2023-02-04 09:00:00 |
| Environmental Engineer   | Green Energy Solutions   | 2023-02-05 14:30:00 |
| Cybersecurity Specialist | CyberSec Guardians       | 2023-02-06 13:00:00 |
| Healthcare IT Consultant | HealthHub Inc.           | 2023-02-07 15:30:00 |
| Manufacturing Engineer   | Manufacturing Innovations | 2023-02-08 16:45:00 |
| EdTech Content Developer | EducationTech Systems    | 2023-02-09 08:15:00 |
| E-commerce Manager       | E-commerce Express       | 2023-02-10 17:15:00 |
+--------------------------+--------------------------+---------------------+
10 rows in set (0.00 sec)
```

;

8. Create an SQL query that calculates and displays the average salary offered by all companies for job listings in the "Jobs" table. Ensure that the query filters out jobs with a salary of zero.

```
mysql> SELECT
    ->     AVG(Salary) AS AverageSalary
    -> FROM
    ->     Jobs
    -> WHERE
    ->     Salary > 0;
+---------------+
| AverageSalary |
+---------------+
|   107000.0000 |
+---------------+
1 row in set (0.00 sec)
```

9. Write an SQL query to identify the company that has posted the most job listings. Display the company name along with the count of job listings they have posted. Handle ties if multiple companies have the same maximum count.

```
mysql> SELECT
    ->     Companies.CompanyName,
    ->     COUNT(Jobs.JobID) AS JobCount
    -> FROM
    ->     Companies
    -> JOIN
    ->     Jobs ON Companies.CompanyID = Jobs.CompanyID
    -> GROUP BY
    ->     Companies.CompanyName
    -> ORDER BY
    ->     JobCount DESC;
+--------------------------+----------+
| CompanyName              | JobCount |
+--------------------------+----------+
| TechCorp                 |        1 |
| FinanceCo                |        1 |
| MedTech Solutions        |        1 |
| Data Innovators          |        1 |
| Green Energy Solutions   |        1 |
| CyberSec Guardians       |        1 |
| HealthHub Inc.           |        1 |
| Manufacturing Innovations |        1 |
| EducationTech Systems    |        1 |
| E-commerce Express       |        1 |
+--------------------------+----------+
10 rows in set (0.00 sec)
```

10. Find the applicants who have applied for positions in companies located in 'CityX' and have at least 3 years of experience.

```
mysql>  SELECT
    ->     Applicants.ApplicantID,
    ->     Applicants.FirstName,
    ->     Applicants.LastName,
    ->     Applicants.Email,
    ->     Applicants.Phone,
    ->     Applicants.Resume
    -> FROM
    ->     Applicants
    -> JOIN
    ->     Applications ON Applicants.ApplicantID = Applications.ApplicantID
    -> JOIN
    ->     Jobs ON Applications.JobID = Jobs.JobID
    -> JOIN
    ->     Companies ON Jobs.CompanyID = Companies.CompanyID
    -> WHERE
    ->     Companies.Location = 'New York'
    ->
    -> ;
+-------------+-----------+----------+-----------------------+--------------+-------------------------+
| ApplicantID | FirstName | LastName | Email                 | Phone        | Resume                  |
+-------------+-----------+----------+-----------------------+--------------+-------------------------+
|           2 | Jane      | Smith    | jane.smith@example.com | +9876543210 | Jane_Smith_Resume.docx |
+-------------+-----------+----------+-----------------------+--------------+-------------------------+
1 row in set (0.00 sec)
```

11. Retrieve a list of distinct job titles with salaries veen $60,000 and $80,000.

```
mysql> SELECT DISTINCT
    ->     JobTitle
    -> FROM
    ->     Jobs
    -> WHERE
    ->     Salary BETWEEN 60000 AND 80000;
Empty set (0.00 sec)
```

12. Find the jobs that have not received any applications least 3 years of experience.

```
mysql> SELECT
    ->     Jobs.JobID,
    ->     Jobs.JobTitle,
    ->     Jobs.JobDescription,
    ->     Jobs.JobLocation,
    ->     Jobs.Salary,
    ->     Jobs.JobType,
    ->     Jobs.PostedDate
    -> FROM
    ->     Jobs
    -> LEFT JOIN
    ->     Applications ON Jobs.JobID = Applications.JobID
    -> LEFT JOIN
    ->     Applicants ON Applications.ApplicantID = Applicants.ApplicantID
    -> WHERE
    ->     Applications.ApplicationID IS NULL
    ->     OR (
    ->         Applicants.ApplicantID IS NOT NULL
    ->         AND NOT EXISTS (
    ->             SELECT 1
    ->             FROM Applicants
    ->             WHERE Applicants.ApplicantID = Applications.ApplicantID
    ->             -- Add logic to check if the applicant has at least 3 years of experience based on the resume or other criteria
    ->             -- For example, you might have keywords in the resume indicating experience
    ->             AND Applicants.Resume LIKE '%experience%'
    ->         )
    ->     );
+-------+-----------------------+-----------------------------------------------------+-----------------+--------+-----------+---------------------+
| JobID | JobTitle              | JobDescription                                      | JobLocation     | Salary | JobType   | PostedDate          |
+-------+-----------------------+-----------------------------------------------------+-----------------+--------+-----------+---------------------+
|   101 | Software Engineer     | Developing cutting-edge software solutions.         | Silicon Valley  | 120000 | Full-time | 2023-01-01 10:00:00 |
|   102 | Financial Analyst     | Analyzing financial data and preparing reports.     | New York        |  90000 | Full-time | 2023-01-02 11:30:00 |
|   103 | Medical Researcher    | Conducting research on innovative medical technologies. | Boston      | 110000 | Full-time | 2023-01-03 12:45:00 |
|   104 | Data Scientist        | Building machine learning models for data analysis. | San Francisco   | 130000 | Full-time | 2023-01-04 09:15:00 |
|   105 | Environmental Engineer | Designing eco-friendly energy solutions.           | Los Angeles     |  95000 | Full-time | 2023-01-05 14:00:00 |
|   106 | Cybersecurity Specialist | Securing networks and systems from cyber threats. | Washington, D.C. | 115000 | Full-time | 2023-01-06 13:20:00 |
|   107 | Healthcare IT Consultant | Implementing technology solutions in healthcare. | Chicago        | 100000 | Full-time | 2023-01-07 15:45:00 |
|   108 | Manufacturing Engineer | Optimizing production processes in manufacturing.  | Detroit         | 105000 | Full-time | 2023-01-08 16:30:00 |
|   109 | EdTech Content Developer | Creating educational content for online learning platforms. | Austin | 85000 | Full-time | 2023-01-09 08:45:00 |
|   110 | E-commerce Manager    | Overseeing online sales and marketing operations.   | Seattle         | 120000 | Full-time | 2023-01-10 17:00:00 |
+-------+-----------------------+-----------------------------------------------------+-----------------+--------+-----------+---------------------+
10 rows in set (0.00 sec)
```

13. Retrieve a list of job applicants along with the companies they have applied to and the positions they have applied for.

```
mysql> SELECT
    ->     Applicants.ApplicantID,
    ->     Applicants.FirstName,
    ->     Applicants.LastName,
    ->     Companies.CompanyName,
    ->     Jobs.JobTitle
    -> FROM
    ->     Applicants
    -> JOIN
    ->     Applications ON Applicants.ApplicantID = Applications.ApplicantID
    -> JOIN
    ->     Jobs ON Applications.JobID = Jobs.JobID
    -> JOIN
    ->     Companies ON Jobs.CompanyID = Companies.CompanyID;
+-------------+-----------+----------+--------------------------+--------------------------+
| ApplicantID | FirstName | LastName | CompanyName              | JobTitle                 |
+-------------+-----------+----------+--------------------------+--------------------------+
|           1 | John      | Doe      | TechCorp                 | Software Engineer        |
|           2 | Jane      | Smith    | FinanceCo                | Financial Analyst        |
|           3 | Michael   | Johnson  | MedTech Solutions        | Medical Researcher       |
|           4 | Emily     | Williams | Data Innovators          | Data Scientist           |
|           5 | Daniel    | Brown    | Green Energy Solutions   | Environmental Engineer   |
|           6 | Sophia    | Miller   | CyberSec Guardians       | Cybersecurity Specialist |
|           7 | Matthew   | Davis    | HealthHub Inc.           | Healthcare IT Consultant |
|           8 | Olivia    | Martinez | Manufacturing Innovations| Manufacturing Engineer   |
|           9 | Christopher| Taylor  | EducationTech Systems    | EdTech Content Developer |
|          10 | Ava       | Jackson  | E-commerce Express       | E-commerce Manager       |
+-------------+-----------+----------+--------------------------+--------------------------+
10 rows in set (0.00 sec)
```

14. Retrieve a list of companies along with the count of jobs they have posted, even if they have not received any applications.

```
mysql> SELECT
    ->     Companies.CompanyID,
    ->     Companies.CompanyName,
    ->     COUNT(Jobs.JobID) AS PostedJobsCount
    -> FROM
    ->     Companies
    -> LEFT JOIN
    ->     Jobs ON Companies.CompanyID = Jobs.CompanyID
    -> LEFT JOIN
    ->     Applications ON Jobs.JobID = Applications.JobID
    -> GROUP BY
    ->     Companies.CompanyID, Companies.CompanyName;
+-----------+--------------------------+-----------------+
| CompanyID | CompanyName              | PostedJobsCount |
+-----------+--------------------------+-----------------+
|         1 | TechCorp                 |               1 |
|         2 | FinanceCo                |               1 |
|         3 | MedTech Solutions        |               1 |
|         4 | Data Innovators          |               1 |
|         5 | Green Energy Solutions   |               1 |
|         6 | CyberSec Guardians       |               1 |
|         7 | HealthHub Inc.           |               1 |
|         8 | Manufacturing Innovations|               1 |
|         9 | EducationTech Systems    |               1 |
|        10 | E-commerce Express       |               1 |
+-----------+--------------------------+-----------------+
10 rows in set (0.00 sec)
```

15. List all applicants along with the companies and positions they have applied for, including those who have not applied.

```
mysql> SELECT
    ->     Applicants.ApplicantID,
    ->     Applicants.FirstName,
    ->     Applicants.LastName,
    ->     Companies.CompanyName,
    ->     Jobs.JobTitle
    -> FROM
    ->     Applicants
    -> CROSS JOIN
    ->     Companies
    -> CROSS JOIN
    ->     Jobs
    -> LEFT JOIN
    ->     Applications ON Applicants.ApplicantID = Applications.ApplicantID AND Jobs.JobID = Applications.JobID;
+-------------+-------------+----------+--------------------------+-------------------------+
| ApplicantID | FirstName   | LastName | CompanyName              | JobTitle                |
+-------------+-------------+----------+--------------------------+-------------------------+
|           1 | John        | Doe      | E-commerce Express       | Software Engineer       |
|           2 | Jane        | Smith    | E-commerce Express       | Software Engineer       |
|           3 | Michael     | Johnson  | E-commerce Express       | Software Engineer       |
|           4 | Emily       | Williams | E-commerce Express       | Software Engineer       |
|           5 | Daniel      | Brown    | E-commerce Express       | Software Engineer       |
|           6 | Sophia      | Miller   | E-commerce Express       | Software Engineer       |
|           7 | Matthew     | Davis    | E-commerce Express       | Software Engineer       |
|           8 | Olivia      | Martinez | E-commerce Express       | Software Engineer       |
|           9 | Christopher | Taylor   | E-commerce Express       | Software Engineer       |
|          10 | Ava         | Jackson  | E-commerce Express       | Software Engineer       |
|           1 | John        | Doe      | EducationTech Systems    | Software Engineer       |
|           2 | Jane        | Smith    | EducationTech Systems    | Software Engineer       |
|           3 | Michael     | Johnson  | EducationTech Systems    | Software Engineer       |
|           4 | Emily       | Williams | EducationTech Systems    | Software Engineer       |
|           5 | Daniel      | Brown    | EducationTech Systems    | Software Engineer       |
|           6 | Sophia      | Miller   | EducationTech Systems    | Software Engineer       |
|           7 | Matthew     | Davis    | EducationTech Systems    | Software Engineer       |
|           8 | Olivia      | Martinez | EducationTech Systems    | Software Engineer       |
|           9 | Christopher | Taylor   | EducationTech Systems    | Software Engineer       |
|          10 | Ava         | Jackson  | EducationTech Systems    | Software Engineer       |
```

16. Find companies that have posted jobs with a salary higher than the average salary of all jobs.

```
mysql> SELECT DISTINCT
    ->     Companies.CompanyID,
    ->     Companies.CompanyName
    -> FROM
    ->     Companies
    -> JOIN
    ->     Jobs ON Companies.CompanyID = Jobs.CompanyID
    -> WHERE
    ->     Jobs.Salary > (SELECT AVG(Salary) FROM Jobs);
+-----------+--------------------+
| CompanyID | CompanyName        |
+-----------+--------------------+
|         1 | TechCorp           |
|         3 | MedTech Solutions  |
|         4 | Data Innovators    |
|         6 | CyberSec Guardians |
|        10 | E-commerce Express |
+-----------+--------------------+
5 rows in set (0.00 sec)
```

17. Display a list of applicants with their names and a concatenated string of their city and state.

```
mysql> SELECT
    ->     ApplicantID,
    ->     FirstName,
    ->     LastName,
    ->     CONCAT(City, ', ', State) AS Location
    -> FROM
    ->     Applicants;
```

18. Retrieve a list of jobs with titles containing either 'Developer' or 'Engineer'.

```
mysql> SELECT
    ->     JobID,
    ->     JobTitle
    -> FROM
    ->     Jobs
    -> WHERE
    ->     JobTitle LIKE '%Developer%' OR JobTitle LIKE '%Engineer%';
+-------+--------------------------+
| JobID | JobTitle                 |
+-------+--------------------------+
|   101 | Software Engineer        |
|   105 | Environmental Engineer   |
|   108 | Manufacturing Engineer   |
|   109 | EdTech Content Developer |
+-------+--------------------------+
4 rows in set (0.00 sec)
```

19. Retrieve a list of applicants and the jobs they have applied for, including those who have not applied and jobs without applicants.

```
mysql> SELECT
    ->     A.ApplicantID,
    ->     A.FirstName,
    ->     A.LastName,
    ->     COALESCE(J.JobTitle, 'No Application') AS AppliedJobTitle
    -> FROM
    ->     Applicants A
    -> LEFT JOIN
    ->     Applications AP ON A.ApplicantID = AP.ApplicantID
    -> LEFT JOIN
    ->     Jobs J ON AP.JobID = J.JobID;
+-------------+-------------+----------+--------------------------+
| ApplicantID | FirstName   | LastName | AppliedJobTitle          |
+-------------+-------------+----------+--------------------------+
|           1 | John        | Doe      | Software Engineer        |
|           2 | Jane        | Smith    | Financial Analyst        |
|           3 | Michael     | Johnson  | Medical Researcher       |
|           4 | Emily       | Williams | Data Scientist           |
|           5 | Daniel      | Brown    | Environmental Engineer   |
|           6 | Sophia      | Miller   | Cybersecurity Specialist |
|           7 | Matthew     | Davis    | Healthcare IT Consultant |
|           8 | Olivia      | Martinez | Manufacturing Engineer   |
|           9 | Christopher | Taylor   | EdTech Content Developer |
|          10 | Ava         | Jackson  | E-commerce Manager       |
+-------------+-------------+----------+--------------------------+
10 rows in set (0.00 sec)
```

20. List all combinations of applicants and companies where the company is in a specific city and the applicant has more than 2 years of experience. For example: city=Chennai.

```
mysql> SELECT
    ->     A.ApplicantID,
    ->     A.FirstName,
    ->     A.LastName,
    ->     C.CompanyName
    -> FROM
    ->     Applicants A
    -> CROSS JOIN
    ->     Companies C
    -> LEFT JOIN
    ->     Applications AP ON A.ApplicantID = AP.ApplicantID
    -> LEFT JOIN
    ->     Jobs J ON AP.JobID = J.JobID AND J.CompanyID = C.CompanyID
    -> WHERE
    ->     C.Location = 'Chennai'
    ->     AND DATEDIFF(YEAR, A.JoiningDate, GETDATE()) > 2;
```